Iterative Solver for Linear System Obtained by Edge Element: Variable Preconditioned Method with Mixed Precision on GPU

Soichiro Ikuno[1], Yuki Kawaguchi[1], Norihisa Fujita[1], Taku Itoh[2], Susumu Nakata[3] and Kota Watanabe[4]
[1]Tokyo University of Technology, 1404-1 Katakura, Hachioji, Tokyo 192-0982, Japan
[3]Seikei University, Musashino, Tokyo 180-8633, Japan
[3]Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan
[4]Hokkaido University, Kita-ku, Sapporo 060-0814, Japan
E-mail ikuno@nal.ikulab.org

***Abstract*—The variable preconditioned iterative solver for linear system obtained by finite element method based on edge element is numerically investigated. For the high performance computing, the method is implemented on Graphics Processing Unit (GPU). The convergence theorem of variable preconditioned iterative solver is guaranteed that the residual equation for the preconditioned procedure can be solved in the range of single precision.**

The results of computations show that mixed precision variable preconditioned GCR method with CR for inner-loop is 5.75 times faster than that of variable preconditioned CG method on CPU.

I. Introduction

Recently, the performance of Graphics Processing Unit (GPU) surpasses that of CPU and various researches of General Purpose for GPU (GPGPU) have been proposed aggressively [1]. Furthermore, GPU architectures have been changed from fixed operation to flexible organization for programmability; therefore, GPUs are capable of scientific computing more than the specific graphics operation. For example, GeForce GTX 285 can perform up to 1063 GFLOPS by using single precision and 88.6 GFLOPS by using double precision. However, there are difficult points in operations using GPU. CUDA is architecture with new hardware and the software to manage the calculation on GPU as a parallel computer developed by the NVIDIA corporation [2]. In addition, when CUDA is used, we do not have to move the data to graphics API. The concept related to the graphics like the texture memory and the frame buffer, etc. did not worry when CUDA is used, and it is possible to treat comparatively with the operation in CPU.

In the discretizing process of magnetostatic problems, the coefficient matrix of the linear system obtained by Finite Element Method (FEM) using edge element becomes singular, symmetric, and large sparse matrix [3]. From this reason, Incomplete Cholesky Conjugate Gradients method is applied for the solver of the linear systems [4]. However, it takes much time to factorize the matrix and it is very difficult to parallelize.

The purpose of the present study is to implement the hybrid scheme of Variable Preconditioned Krylov subspace method that uses single precision and double precision operations on GPU using CUDA and to solve the linear system obtain from FEM with edge element by means of the code.

Let $\boldsymbol{x}_0$ be an initial guess.
Set $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
roughly solve $A\boldsymbol{z}_0 = \boldsymbol{r}_0$ using some iterative method
Set $\boldsymbol{p}_0 = \boldsymbol{z}_0$
for $k = 0, 1, \cdots$, until $\|\boldsymbol{r}_k\|_2 / \|\boldsymbol{b}\|_2 \leq \varepsilon$ **do**
  $\alpha_k = \dfrac{(\boldsymbol{r}_k, \boldsymbol{z}_k)}{(\boldsymbol{p}_k, A\boldsymbol{p}_k)}$
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
  $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$
  roughly solve $A\boldsymbol{z}_{k+1} = \boldsymbol{r}_{k+1}$ using some iterative method
  $\beta_k = \dfrac{(\boldsymbol{r}_{k+1}, \boldsymbol{z}_{k+1})}{(\boldsymbol{r}_k, \boldsymbol{z}_k)}$
  $\boldsymbol{p}_{k+1} = \boldsymbol{z}_{k+1} + \beta_k \boldsymbol{p}_k$
end for

Fig. 1. The algorithm of variable preconditioned conjugate gradient (VPCG) method.

II. Variable Preconditioned Krylov Subspace Method

It is well known that a preconditioning strategy can improve the performance for solving a linear system $A\boldsymbol{x} = \boldsymbol{b}$ using the Krylov subspace method and various preconditioning strategies have been developed and numerically investigated [5][6][7][8]. Here, A, $\boldsymbol{x}$ and $\boldsymbol{b}$ denote a coefficient matrix, an unknown vector, and a known vector, respectively. Generally, a preconditioned matrix M is determined by incomplete LU decomposition, and a vector $M^{-1}\boldsymbol{r}_k$ is calculated at k-th iteration by using backward substitution or incomplete Cholesky factorization. Here, $\boldsymbol{r}_k$ denotes residual vector at k-th iteration. The calculation time of solving linear system is relatively large for the preconditioned part.

K. Abe *et al.* developed new preconditioning strategy which is called the variable preconditioning method [9]. VPGCR has two nested iterations for GCR and variable preconditioning for GCR are called as outer-loop and inner-loop, respectively. In the preconditioned procedure, the residual equation is solved to determine the preconditioner for the outer-loop. The algorithm of VPGCR can be extended for symmetric matrix solver. The algorithm of variable preconditioned conjugate method is shown in Fig. 1.

The convergence theorem of variable preconditioned Krylov subspace method is guaranteed that the residual of the method converges if the relative residual norm of inner-loop satisfies

the following inequality.

$$r_{\rm in} = \frac{||\boldsymbol{r}_{k+1} - A\boldsymbol{z}_{k+1}||_2}{||\boldsymbol{r}_{k+1}||_2} < 1. \quad (1)$$

Here, $||\boldsymbol{x}||_2$ denotes 2-norm of vector $\boldsymbol{x}$. That is to say; the residual equation can be solved roughly by using iterative method with only a few iteration. Generally, a stationary iterative method such as Gauss-Seidel method, is adopted for variable preconditioning procedure.

To elicit the high performance of GPU, it is necessary to use single precision calculations in a meaningful way because GPU is developed as a device used for drawing of the graphics. Additionally, the convergence theorem of variable preconditioned Krylov subspace method is guaranteed that the residual equation can be solved in the range of single precision, which means that the method is applicable method to elicit the high performance of GPU. Therefore, variable preconditioned Krylov subspace method with mixed precision that uses single precision operation for inner-loop and double precision operation for outer-loop is adopted for the solver on GPU and the method is called the mixed precision method.

III. NUMERICAL EVALUATION

In this study, the Problem 20 in Testing Electromagnetic Analysis Methods (T.E.A.M) Workshop is employed for the benchmark. The analytic region is divied into four symmetric region, and a piece of the region is adopted for calculation. The governing equation:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \boldsymbol{A} \right) = \boldsymbol{J}, \quad (2)$$

is discretized by Finite Element Method (FEM) with edge element. Here, μ, $\boldsymbol{A}$ and $\boldsymbol{J}$ denote a magnetic permeability, a vector potential, and a current density, respectively. The size of the analytic domain which is including air region is $0 \leq x \leq 0.15$, $0 \leq y \leq 0.15$, and $-0.05 \leq z \leq 0.20$. Moreover, value of number of node $N_{\rm node}$, number of element $N_{\rm elem}$, number of edge $N_{\rm edge}$ and degree of freedom N is $N_{\rm node} = 98105$, $N_{\rm elem} = 535898$, $N_{\rm edge} = 649325$, $N = 603356$. Essentially, original problem of T.E.A.M Workshop problem 20 is a nonlinear magnetostatic field model. For the simplicity, value of relative magnetic permeability is fixed as 200, and the problem becomes a linear problem. In addition, coil current is 1000 [A turns]. The evaluation environment is shown in Table I. Note that, the calculations on CPU are parallelized in four threads by OpenMP and SIMD (Single Instruction Multiple data) operation is automatically adopted on each thread by using compiler option `-O3`.

We investigate the values CPU time and speedup for various variable preconditioned Krylov subspace method, and the results are shown in Table II. The termination condition for inner-loop $\varepsilon_{\rm in}$ and for outer-loop $\varepsilon_{\rm out}$ is fixed as $\varepsilon_{\rm in} = 10^{-3}$, $\varepsilon_{\rm out} = 10^{-8}$, respectively. And, VPCG method on CPU with CG for inner-loop is adopted for the origin of speedup. We can see from this table the mixed precision methods on GPU are faster than that of CPU or standard method on GPU in both cases. Especially, VPGCR method with CR for inner-loop is 5.75 times faster than that of VPCG method on CPU.

TABLE I
EVALUATION ENVIRONMENT.

OS	Ubuntu Linux 9.10
CPU	Intel Core i7 930
CPU memory	12 GB
CPU compiler	gcc 4.5.0
Compiler option	`-O3 -march=native -free-vectorize -fopenmp`
GPU	GeForce GTX 480
GPU memory	1.5 GB
GPU compiler	nvcc 3.1
Compiler option	`-O -arch=sm_20`

TABLE II
THE RESULTS OF EVALUATIONS. HERE, $N_{\rm max}$, $\tau_{\rm CPU}$ AND M DENOTE MAXIMUM ITERATION NUMBER FOR INNER-LOOP, CPU TIME AND NUMBER OF OUTER ITERATION NUMBER, RESPECTIVELY. THE RESTART PARAMETER FOR GCR AND GMRES IS FIXED AS 100. FURTHERMORE, THE NAME OF SOLVER IN BRACKET IN SECOND COLUMN DENOTES ADOPTED SOLVER FOR INNER-LOOP.

PU	Method	$N_{\rm max}$	$\tau_{\rm CPU}$ [sec]	M	Speedup
CPU	VPCG (CG)	600	201.53	18	1.0
	VPGCR (CR)	300	173.65	27	1.16
	VPGMRES (CR)	350	182.43	24	1.10
GPU	VPCG (CG)	600	53.46	18	3.76
	VPGCR (CR)	300	43.98	27	4.58
	VPGMRES (CR)	400	45.37	21	4.44
GPU (Mixed)	VPCG (CG)	600	41.81	19	4.82
	VPGCR (CR)	200	35.01	43	5.75
	VPGMRES (CR)	350	35.07	25	5.74

It is notable that VPGCR and VPGMRES are much faster than VPCG. The number of operations of GCR or GMRES is larger than that of CG because GCR and GMRES are solver for asymmetric matrix. This is because both methods are developed based on Arnoldi process. Therefore, the residual history is very smooth like a stationary iterative method.

REFERENCES

[1] `http://gpgpu.org/`
[2] `http://www.nvidia.com/object/cuda_develop.html`
[3] H. Igarashi, *On the Property of the CurlCurl Matrix in FiniteElement Analysis With Edge Elements*, IEEE Trans. Magn., Vol. 37, No. 5 (2001) 3129-3132.
[4] K. Fujiwara, T. Nakata, H. Fusayasu, *Acceleration of convergence characteristic of the ICCG method*, IEEE Trans. Magn., Vol. 29 No. 2 (1993) pp.1958-1961.
[5] H. Igarashi, T. Honma, *Convergence of Preconditioned Conjugate Gradient Method Applied to Driven Microwave Problems*, IEEE Trans. Magn., Vol. 39, No. 3 (2003) 1705-1708.
[6] T. Mifune, T. Iwashita, M. Shimasaki, *A Novel Algebraic Multigrid Preconditioning for Large-Scale Edge-Element Analyses*, IEEE Trans. Magn., Vol. 43, No. 4 (2007) 1481-1484.
[7] H. Igarashi, N. Yamamoto, *Effect of Preconditioning in Edge-Based Finite-Element Method*, IEEE Trans. Magn., Vol. 44, No. 6 (2008) 942-945.
[8] H. Igarashi, N. Yamamoto, *Folded Preconditioner: A New Class of Preconditioners for Krylov Subspace Methods to Solve Redundancy-Reduced Linear Systems of Equations*, IEEE Trans. M agn., Vol. 45, No. 5 (2009) 2068-2071.
[9] K. Abe, S. L. Zhang, *A variable preconditioning using the SOR method for GCR-like methods*, Int. J. Numer. Anal. Model. 2, no. 2, pp. 147–161, 2005.
[10] A. Saitoh, A. Kamitani, *GMRES With New Preconditioning for Solving BEM-Type Linear System*, IEEE Trans. Magn., Vol. 40, No. 2 (2004) 1084-1087.